(Kruskal)

\[G = (V, E), w>\]

1. ב-\(B\) \(\leftarrow\) \(\emptyset\), \(C\) \(\leftarrow\) \(E\)
2. כל ת UIL \(V, B\) \(\rightarrow\) \(E, C\)
3. \(B\) \(\leftarrow\) \(B \cup \{e\}\)
4. \(B\) \(\rightarrow\) \(V\) \(\rightarrow\)

(Prim)

\[G = (V, E), w>\]

1. בחר \(r \in V\) \(\rightarrow\)
2. \(B\) \(\leftarrow\) \(\emptyset\), \(C\) \(\leftarrow\) \(E\), \(S\) \(\leftarrow\) \(\{r\}\)
3. \(|B| < |V| - 1\)
4. \(u \in S, v \notin S\) \(e \in E\) \(\rightarrow\)
5. \(B\) \(\leftarrow\) \(B \cup \{e\}\)
6. \(S\) \(\leftarrow\) \(S \cup \{v\}\)
7. \(B\) \(\rightarrow\) \(V\) \(\rightarrow\)

**Initialize-Single-Source** \((G, s)\)

1. \text{for each vertex } v \in V[G] \text{ do}
2. \(d[v] \leftarrow \infty\)
3. \(\pi[v] \leftarrow \text{NIL}\)
4. \(d[s] \leftarrow 0\)

**Relax** \((u, v, w)\)

1. if \(d[v] > d[u] + w(u, v)\) then
2. \(d[v] \leftarrow d[u] + w(u, v)\)
3. \(\pi[v] \leftarrow u\)

**Dijkstra** \((G, w, s)\)

1. **Initialize-Single-Source** \((G, s)\)
2. \(S \leftarrow \emptyset\)
3. \(Q \leftarrow V[G]\)
4. \text{while } Q \neq \emptyset \text{ do}
5. \(u \leftarrow \text{Extract-Min}(Q)\)
6. \(S \leftarrow S \cup \{u\}\)
7. \text{for each vertex } v \text{ in } Q \text{ such that } v \in Adj[u] \text{ do}
8. \(\text{Relax}(u, v, w)\)
DFS(G)
1 For each vertex \( u \in V[G] \) do
2 \( \text{color}[u] \leftarrow \text{WHITE} \)
3 \( \pi[u] \leftarrow \text{NIL} \)
4 \( \text{time} \leftarrow 0 \)
5 For each vertex \( u \in V[G] \)
6 If \( \text{color}[u] = \text{WHITE} \) then
7 \( \text{DFS-VISIT}(u) \)

DFS-VISIT(u)
1 \( \text{color}[u] \leftarrow \text{GRAY} \)
2 \( d[u] \leftarrow \text{time}; \text{time} \leftarrow \text{time} + 1 \)
3 For each vertex \( v \in \text{Adj}[u] \)
4 If \( \text{color}[v] = \text{WHITE} \) then
5 \( \pi[v] \leftarrow u \)
6 \( \text{DFS-VISIT}(v) \)
7 \( \text{color}[u] \leftarrow \text{BLACK} \)
8 \( f[u] \leftarrow \text{time}; \text{time} \leftarrow \text{time} + 1 \)

BELLMAN-FORD(G,u,v)
1 Initialize-Single-Source(G,s)
2 for \( i \leftarrow 0 \) to \( |V[G]| - 1 \) do
3 for each edge \((u,v) \in E[G]\) do
4 Relax(u,v,w)
5 for each edge \((u,v) \in E[G]\) do
6 if \( d[v] > d[u] + \omega(u,v) \) then
7 return FALSE
8 return TRUE

FLOYD_WARSHALL(W)
1 \( n \leftarrow \text{rowa}[W] \)
2 \( D \leftarrow W; \)
3 for \( k \leftarrow 1 \) to \( n \) do
4 for \( i \leftarrow 1 \) to \( n \) do
5 for \( j \leftarrow 1 \) to \( n \) do
6 \( d_{ij} \leftarrow \min\{d_{ij},d_{ik} + d_{kj}\} \)
7 Return D

FOROUGH-METHOD(G,c,s,t)
1 initializes flow \( f \) to 0
2 while there exists an augmenting path \( p \) in \( G_f \) do
3 \( c_f(p) = \min\{c_f(u,v) : (u,v) \text{ is in } p\} \)
4 augment flow \( f \) along \( p \) by \( c_f(p) \)
5 return \( f \)
Dinitz \((G, c, s, t)\)
1. \(f \leftarrow 0\);
2. construct the residual network \(N_f = (G_f, c_f, s, t)\);
3. \(\text{while there is a path from } s \text{ to } t \text{ in } G_f \text{ do.}\)
4. construct the layered network \(L_f = (L_f, c_f, s, t)\);
5. find a blocking flow \(b\) for \(L_f\);
6. \(f \leftarrow f + b\);
7. construct the residual network \(N_f\);

LayeredNetwork_Construction \((N_f)\)
1. \(V_0 \leftarrow \{s\}; \quad i \leftarrow 0;\)
2. \(\text{while } (V_i \neq \phi) \text{ and } (t \not\in V_i) \text{ do}\)
3. \(V_{i+1} \leftarrow \phi; \quad E_{i+1} \leftarrow \phi;\)
4. for each \(u \in V_i\) do
5. \(\text{for each } v \in V \text{ such that } (u, v) \in E_j \text{ and } (v \not\in V_j, \forall j \leq i) \text{ do}\)
6. \(\text{if } (v \not\in V_{i+1}) \text{ then}\)
7. add \(v\) to \(V_{i+1}\);
8. add \(\langle u, v \rangle\) to \(E_{i+1}\);
9. \(i \leftarrow i + 1;\)
10. if \(V_i = \phi\) then
11. return \(L_f = (\phi, c_f, s, t)\)
12. \(L_f \leftarrow (V_0 \cup ... \cup V_i, E_1 \cup ... \cup E_i)\);
13. return \(L_f = (L_f, c_f, s, t)\);

BlockingFlow \((\mathcal{L}_f)\)
1. \(b \leftarrow 0\)
2. repeat
3. find a path \(p\) from \(s\) to \(t\) in \(L_f\); let its bottleneck capacity be \(c_r(p)\)
4. \(\text{for each } \langle u, v \rangle \in p \text{ do}\)
5. \(b(u, v) \leftarrow b(u, v) + c_r(p)\)
6. \(b(v, u) \leftarrow -b(u, v)\)
7. \(c_r(u, v) \leftarrow c_r(u, v) - c_r(p)\)
8. \(\text{if } c_r(u, v) = 0 \text{ then}\)
9. remove \(\langle u, v \rangle\) from \(L_f\)
10. if \(\text{indegree}(v) = 0\) then
11. CleanForward \((v)\)
12. until there is no path from \(s\) to \(t\) in \(L_f\)
CleanForward \((u)\)

1. if \((u,v) \neq t\) then
2. for all \(v\) such that \(\langle u,v \rangle \in E[L_f]\) do
3. remove \(\langle u,v \rangle\) from \(E[L_f]\)
4. if \(\text{indegree}(v) = 0\) then
5. CleanForward \((v)\)
NP-Complete Problems:

Definition:

CNF formula is a boolean formula where between the clauses there is an AND and each clause is an OR of literals.

Question:

Is there a assignment to the variables that satisfies the formula?

CNF-SAT

Definition:

CNF formula in 3-CNF (each clause has 3 literals).

Question:

Is there an assignment to the variables that satisfies the formula?

NAE-K-SAT

Definition:

CNF formula in k-CNF (k literals in each clause).

Question:

Is there an assignment to the variables that satisfies the formula, such that every clause has at least one literal true and at least one literal false?

Clique

Definition:

Undirected graph G, and number natural k.

Question:

Is there a set of k vertices such that between every two vertices in the set there is an edge?

Independent-Set

Definition:

Undirected graph G = (V, E), and number natural k.

Question:

Is there a set of k vertices such that between every two vertices in the set there is no edge?

Vertex-Cover

Definition:

Undirected graph G = (V, E), and number natural k, and C ⊆ V. For every edge (u, v) ∈ E, u ∈ C or v ∈ C.

Question:

Is there a set of k vertices that cover all the edges?
Set-Cover

\[ U = A_1 \cup A_2 \cup \ldots \cup A_n, \quad A = \{ A_1, A_2, \ldots, A_n \} \]

\[ \text{Question: } B = \bigcup_{b \in B} b \text{ is a subset of } A, k \leq |B|, \quad \text{find a set } A \subseteq A \text{ such that } |A| \leq k \]

KnapSack

\[ \{ w_1, w_2, \ldots, w_n \}, \{ a_1, a_2, \ldots, a_n \} \]

\[ W, P, \{ p_1, p_2, \ldots, p_n \} \]

\[ \text{Question: } \sum_{i \in A} a_i \cdot p_i \leq P - \sum_{i \in B} w_i \leq W \]

Ham-Path

\[ G = (V, E) \]

\[ s, t \in V \]

\[ \text{Question: } \text{does } G \text{ contain a simple path from } s \text{ to } t? \]

Ham-Cycle

\[ G = (V, E) \]

\[ \text{Question: } \text{does } G \text{ contain a cycle that visits each vertex exactly once?} \]

Partition

\[ A = \{ a_1, a_2, \ldots, a_n \} \]

\[ \text{Question: } \text{is } \sum_{a \in A} a = \sum_{b \in B} b, \text{ and } |A| + |B| = n? \]

Subset-Sum

\[ T, A = \{ a_1, a_2, \ldots, a_n \} \]

\[ \text{Question: } \sum_{b \in B} b = T, \text{ and } |A| + |B| = n? \]

3-Coloring

\[ G = (V, E) \]

\[ (u, v) \in E \]

\[ \text{Question: } \text{is } G \text{ 3-colorable?} \]